Chapter 12 SBS616 Module

SBS616 Module

In 1996, Science Applications International Corporation delivered and installed a five-axis ELECTRO-LOAD system for NAVAIR 4.3.2.4. Bihrle Applied Research was tasked to develop an interface for the BAR D-Six simulation environment to serve as a flight simulation host for the NAVAIR system. Under this tasking, Bihrle Applied Research developed an exertion to an existing D-SIX Hardware interface module, D-SIX IOD. This document provides details pertaining to the development and implementation of this extension.

NAVAIR ELECTRO-LOAD System and Set Up

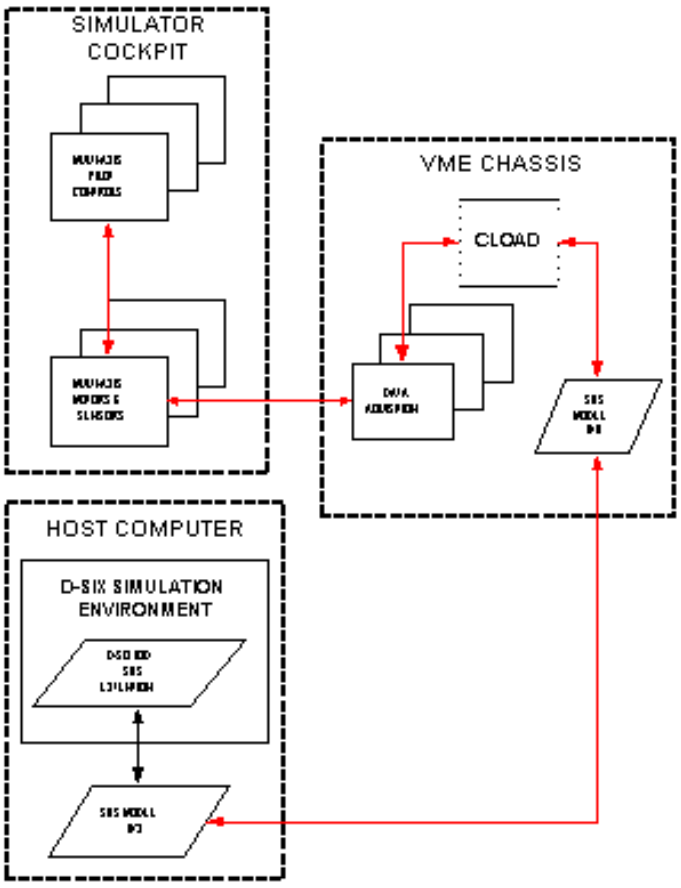

Figure 1. Diagram of the NAVAIR ELECTRO-LOAD system setup.

The ELECTRO-LOAD system delivered to NAVAIR 4.3.2.4 is a five-axis control loading system and was installed in an existing simulated cockpit provided by the US Navy. The control loading system consists of mechanical linkages to the stick, rudder pedals, and throttle; force, position, and acceleration sensors; torque motors; a VME computer chassis with a Pentium Processor running MS DOS v6.22; data acquisition cards; shared memory; and control loading software, CLOAD.

The CLOAD software is run on the ELECTRO-LOAD Pentium communicates with data acquisition hardware and shared memory. See Figure 1. Data acquisition hardware on the VME chassis include individual data acquisition cards for each loaded axis where all measurement data and loading commands are transmitted to and from CLOAD. The shared memory provided by an SBS Model 616 card is used by CLOAD to communicate with a HOST computer. The

HOST computer runs flight simulation software that interfaces with the ELECTRO-LOAD SYSTEM. Through shared memory, the HOST receives pilot control state and observation data. In turn it may provide control-loading model parameter information. There are three basic control-loading models modes that may be run CLOAD, Single-Mass, Double-Mass, and Host. The Single and Double-Mass models are implemented in the CLOAD software and allow modification through parameter data files and HOST provided parameters through shared memory. In the Host mode, control laws for the loading system are implemented on a HOST computer thus providing total control over the control loading system. For more information on the model modes, refer to the ELECTRO-LOAD Software Operations Manual (Reference 1). For the task at hand, a Pentium PC running Windows 98 and the BAR D-Six reconfigurable simulation environment served as the HOST and communicated with the VME based shared memory using an SBS Model 673 PCI card. Figure 1 contains the NAVAIR HOST/ELECTRO-LOAD schematic.
Additional information on the system set up is available in Appendix A.

SBS616 Prevention and Troubleshooting

There are several issues that may cause problems with the use of the SBS 616 module.

1. Always check variable settings.

When variables for input or output do not seem to be working correctly, remember to check the memory location and the unit type. Incorrect memory locations for reading or writing of variables could cause unknown, or even dangerous problems. The SAIC User Interface specification must be followed to insure the correct memory location of variables. Also, variable unit assignments can cause unknown problems. For instance, if the variable being written to is a long variable, but it is set up as a float variable in the SBS 616 configuration window, then a floating point number will be written to the variable. This will either cause the variable to not work or even cause serious problems with the force loader. Remember to check these variables for memory accuracy and for unit assignment.

2. Always make sure VME is running before running D-Six.

It is important to note that D-Six must have the PCI to VME card system working BEFORE D-Six is running. If the system is not up, then D-Six will either not load the SBS 616 module or could even lock the PC system. This problem is due to the initialization of the SBS card drivers. If the card drivers return an error due to the cards not being active, D-Six will try to handle the error, but error handling in this instance is not confirmed. This problem could be compounded by running D-Six after an error is thrown. In this instance, the I/O Driver will still try to write to a non-existent variable and increase the chances of D-Six and the PC locking.

3. Do Not reboot VME system while D-Six is running.

If the VME system is rebooted while D-Six is running, D-Six WILL LOCK UP. This is a total PC lock up and will crash your system. DO NOT reboot VME while D-Six is running. To Reboot the Control Load System exit from D-Six first then proceed.

4. Use MEMCHECK.EXE/CLEARMEM.EXE on VME side to confirm variables are writing or reading correctly.

If needed, there is a program called MEMCHECK.EXE in the cload directory. Memcheck allows the User to visually inspect the shared memory locations and what is being written or read from them. D-Six can run correctly while Memcheck is running and will show the User which memory is being read to or written from.
CLEARMEM.EXE clears the shared memory buffer and should be used prior to running CHECKMEM when examining HOST input to shared memory.

5. Use BTQCHECK.EXE on PC side to confirm variables are writing or reading correctly.

BTQCHECK.EXE is an SBS utility on the HOST computer that allows the User to watch and test the shared memory. This is similar to MEMCHECK.EXE above. Remember to correctly set the type of memory to be watched and the location of the base memory to the correct values.

6. Make sure SHMEM.DAT is correctly configured.
SHMEM.DAT is the file that controls the shared memory settings that cload uses. If this file is incorrectly set, then cload will not properly use the shared memory. The correct SHMEM.DAT file is shown in Appendix B. The main settings in the SHMEM.DAT file are the shared_memory_type and shared_memory_bus_address. Shared_memory_type must be set to "pvme_bit3" which is the VME SBS card and shared_memory_bus_address must be set to 0x0909000. The 0x0909000 has two components: 0x900000 which is the location of shared memory on the SBS VME card and 0x9000 which is a status register. Make sure that the location of shared memory, 0x900000, is correctly set.
7. Make sure that you are using Dual Port Memory.
The SBS cards are currently set to use Dual Port memory. This means that they are accessing shared memory that is located in the Dual Port slot on the VME SBS 616 card. The SBS 616 card should be set up to the memory location of the Dual Port memory of 0x900000 which is the normal cload VME memory address. If the card is not set up to this memory address location, the shared memory will not function correctly.
8. Make sure the PCI Card settings are correct.
On the PCI SBS 973 card, the following settings must be entered. These can be entered by opening B3SetDef. After B3SetDef is opened, select the adapter you want under Adapter Parameters and click on Set Adapter Parameters.

Local Memory 0x80000
Remote RAM 0x900000
Remote Window Size 0x80000
Check Use Dual Port Memory.

All other setting can be left alone.

References
1. ELECTRO-LOAD Software Operations Manual, SAIC, 1996.
2. D-SIX I/O Interface Manual, Bihrle Applied Research Inc. 1999.
3. Model 973 Support Software Manual.

SBS16 D-Six IOD Module Extension

D-SIX IOD
D-SIX IOD is an interface for input and output to and from the D-SIX simulation environment and can be accessed under the TOOLS menu in the D-SIX control panel. Through D-SIX IOD the user has access to all simulation variables as well as variables associated with particular hardware devices. Through a graphical interface hardware parameters may be configured to interact with simulation variables, for example control stick buttons can be configured to trigger simulation events. D-SIX IOD also provides the ability to save input and output device configuration information. Additional information about IOD is available in D-SIX I/O Interface Manual (Reference 2).
SBS Extension
To provide D-SIX IOD with appropriate data from the ELECTRO-LOAD system an extension for the communication between the SBS 673 and SBS 616 board was developed. This extension automates the configuration of the SBS 673 board for use within the D-SIX simulation environment and also provides all the functionality needed to correct work with the PCI interface. The extension also provides a graphical interface to define variables and access the shared memory on the SBS 616 board.

Using the SBS Extension - STEP 1 LOAD

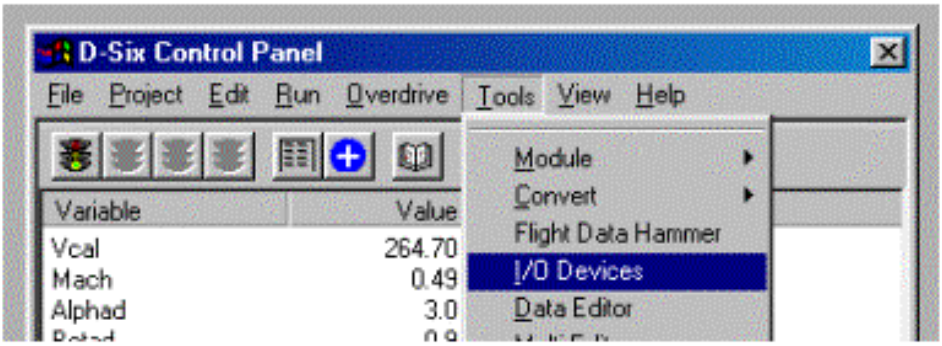

Figure 2. I/O Devices selection.

If the SBS 616 module is not loaded no data will be available to the HOST computer in D-SIX. To check status of the SBS 616 extension, select the I/O Devices option under the tools menu, Figure 2.

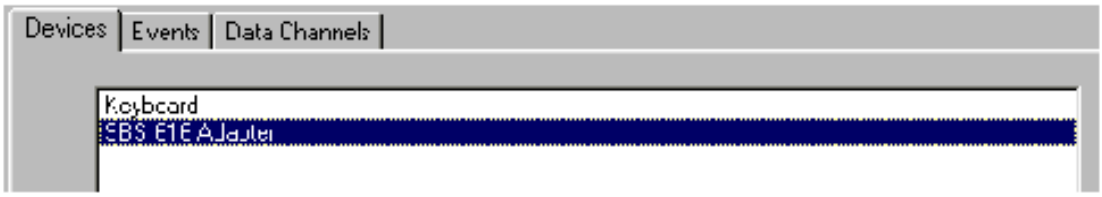

Figure 3. IOD device list.

If the SBS extension is loaded, “SBS 616 Adapter” will appear in the device list, Figure 3.

If the SBS 616 Adapter is not loaded, load a D-SIX module using the following steps:
1. IMPORTANT: Power up the ELECTRO-LOAD SYSTEM and run the CLOAD program. See ELECTRO LOAD USERS GUIDE.
2. From the Modules.ini file insert the name SBS616.d6x.
3. Start D-Six

If the module has been previously loaded and used, it will find all the saved information on the previous configuration of the board. When the SBS 616 module is first used, then no information will be configured on the board. If the module is unable to load correctly, a message box will appear with an error "bt_open(): ERROR: <<error message>>". See Troubleshooting Section for more information on errors.

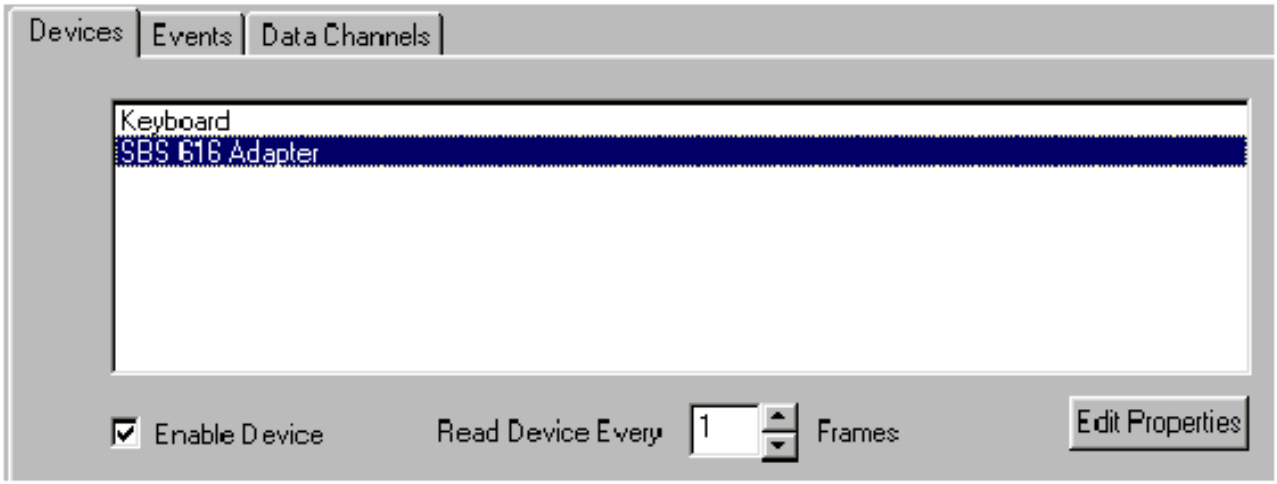

Figure 5. Device list in IOD.

Once loaded, the card will show up under the I/O Device window (Devices tab) as "SBS 616 Adapter". To edit the configuration of the SBS 616 Adapter, click on Edit Properties, Figure 5. To Close the SBS 616 module, select TOOLS>MODULES>UNLOAD MODULES. Then choose the SBS616.d6x module and select OK.

STEP 2 SBS Configuration Utility

The SBS 616 Configuration Utility, Figure 6, allows the User to dynamically modify the I/O Device Interface and can be accessed by editing the properties of the SBS 616 device under the DEVICES tab in the I/O Devices Module. These SBS settings will affect all D-Six projects that are loaded with the SBS module.
The following section provides a description of the items

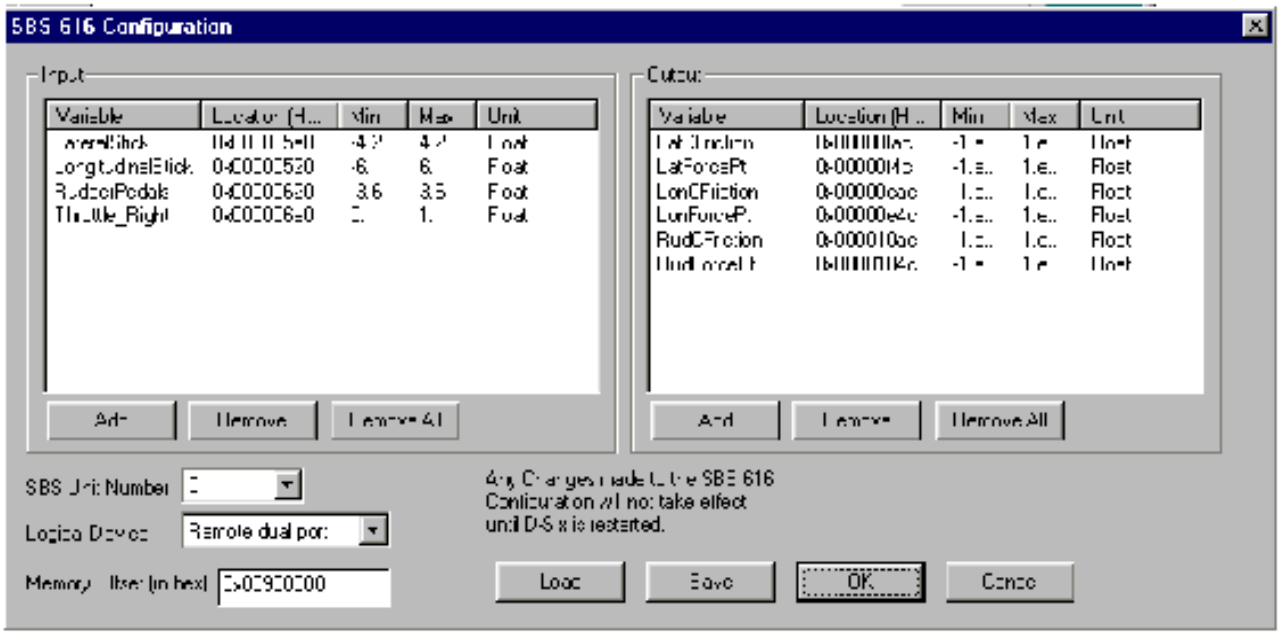

Figure 6. SBS Configuration Utility

SBS Unit Number. Each SBS card within the PC has a Unit Number associated with it. The number is configured by the SBS configuration utility B3SetDef.exe. For more information on the Unit Number, see the PDF manual "Model 973 Support Software Manual" (Reference 3).

Logical Device. The logical device is the type of memory that the module is interacting with. The possible values and a short description of each is:
Remote Dual Port. This is memory that is located on the SBS 616 card as an add on RAM card. This is the usual method of accessing shared memory.
Remote I/O (A16). This is I/O with a 16 bit addressing space. Memory located on PC.
Remote RAM (A24). This is Remote RAM addressed in 24 bit space. Memory is located on the PC as normal RAM that can be accessed by both PCI and VME card.
Remote RAM (A32). This is Remote RAM addressed in 32 bit space. Memory is located on the PC as normal RAM that can be accessed by both PCI and VME card.
Local Memory. Local memory is normal RAM memory located on the PC and only accessed by the PCI card.
Default Device. Default Logical Device. Usually Remote Dual Port.

Memory Offset (in hex). This is the hexadecimal representation of the memory offset into the Logical Device memory type. For Remote RAMs and Dual Port RAM this number will usually be a actual memory address value. For Local Memory this value will be 0x0. Whenever a variable is read or written to the shared memory, the values memory address will be added to the Memory Offset to get an actual memory location to access.
This offset is set by jumpers on the SBS 616 card. See the SBS 616 Manual for more information on how to set this. As per the SCT cockpit for NAVAIR, this value is 0x00900000 for a Remote dual port Logical Device.

Input / Output. The two grouped boxes are Input and Output variables. These are lists of variables that will be passed to and from the shared memory. These variables appear under Data Channels in the I/O Device window where they are mapped to D-SIX simulation variables. The variables are grouped into those that are read from the shared memory (Input) and those that are written to the shared memory (Output). For purposes of timing, Input variables are read first and the Output variables are written.
ELECTRO-LOAD documentation has a complete list of variables read to and from shared memory during the execution of single and double mass model modes, as well as host model mode. The user is encouraged to examine these lists thoroughly.
The Input / Output variables can be edited by double clicking on a variable in the list box. A Variable Settings window will appear, Figure 7, and allow the user to modify variable properties as required. These settings are described below.

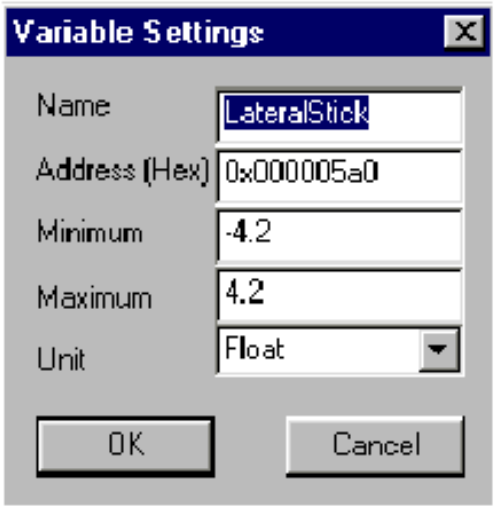

Figure 7. Variable properties.

Variable The name of the variable as it will show up in I/O Devices Data Channels.

Location(Hex) - The memory location of the variables where 0x0 is the start of the shared memory. This is in hexadecimal.
Minimum - The minimum value the variables will have. If the variable to read or write is below this value, then the value will be changed to the minimum.
Maximum - The maximum value the variables will have. If the variable to read or write is above this value, then the variable will be changed to the maximum.
Unit - The Unit that the variable data is read and written in. Possible values are float, long, and unsigned long. Variables are read from memory and translated into this Unit. As such, it will be possible for a floating-point number to be read as a long value and subsequently have a loss of data resolution or just bad data. Specific variable information is provided in the ELECTRO-LOAD documentation.
New Input / Output variables can be added by clicking on the Add button. This will add a new variable called "New Variable : #" where number increments. New variables are floating point variables with a min and max of -10000 and 10000. New variables initially point at 0x0.
To remove variables that are no longer needed, highlight the variable and press the Remove button. To remove all current Input or Output variables press the Remove All button. Note: both these buttons will permanently remove the variables.

Load and **Save.** These two buttons allow the variable configuration to be saved or loaded to and from a file in the users Project directory. SBS 616 configuration files have a "sbs" extension on them.
OK and **Cancel**. To save changes to the SBS configuration click OK. To cancel changes, click Cancel. NOTE: Changes will not take effect until D-Six has been restarted.
STEP 3 Restart D-SIX
In order for SBS extension changes made in the configuration utility to take effect, D-SIX must be restarted prior to operation. This does not included changes made to the IOD events or data channel configuration.

Figure 8. SBS Enable Selection

STEP 4 Enable/Disable Variables Being Used
Each SBS variable can be enabled and disabled individually based on a per project basis. This allows the User to use variables for some projects that affect the SBS system, while making sure that in other projects the variable is ignored or kept constant. To enable or disable individual variables, open the SBS 616 Enabler.
The Enabler can be opened from TOOLS->SBS ENABLE when the SBS module is loaded (see Figure 8). When the SBS 616 Enabler is active, it will allow each individual variable to be enabled or disabled (see Figure 9). By double clicking on the individual variable, the status of the variable will change from Enabled to Disabled and vice versa.

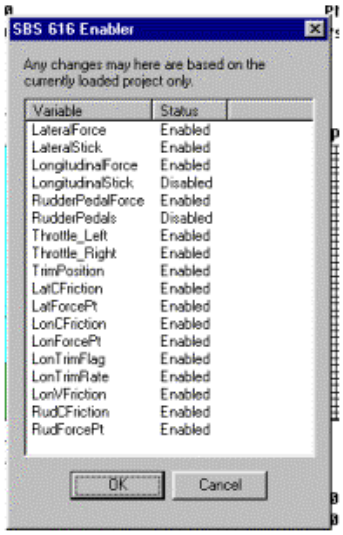

Figure 9. SBS 616 Enabler

When an individual variable is disabled, **the variable will not be written or read from the SBS 616 card**. This means that a variable will not change anything on the SBS card or it will not change the variable that is has been assigned to within IOD.

Jumper Settings for the SBS 616 card

The following jumpers on the SBS 616 card must be set:

System Jumpers: leave as default.

Bus Grant and Bus Request Jumpers (BGO-BGI): This is the setting for when VME card is not the system controller. (pg. 108 in SBS 616 manual).

Set 0, 1, 2 to middle position jumped.
Set 3 to both jumpers in.
Set BR 3 in.

I/O Range Jumpers: This should be set so that only the A15 high/low jumpers are out. All other jumpers should be in.

Remote RAM jumpers (REM-RAM): The settings for these must be disable. Remove A32 and A24 so that REM-RAM is not used.

Dual Port RAM Jumpers: This should be set so that the memory address if 0x900000 to 0x980000. Also this should be set to 24-bit space (A24 in).

23 High/Low Out
20 High/Low Out
19 High Out
A32 out.
All others in.

Transmitted Interrupt Jumpers: Default.

Received Interrupt Jumpers: All jumpers out.

Address Bias Jumpers: Default.

SBS616 File Listing of SHEM.DAT

```
/*****************************************************************************
          File:  shmem.dat
          Purpose: Shared memory address and control registers.
                    Not all parameters are used be each shared memory board.
>>>>   Consult the manuals carefully before altering   <<<<
>>>>   any of these parameters. Any changes from the delivered <<<<
>>>>   configuration could adversely affect the performance and <<<<
>>>>   safety of the ELECTRO-LOAD system.           <<<<
*****************************************************************************/
shared_memory_type     /* Type of shared memory card being used */
pvme_bit3
shared_memory_bus_address  /* Address on computer bus for shared mem card */
0x909000
shared_memory_base_address  /* Base address for start of shared data [0x...] */
0xE0000
memory_size          /* Size of shared memory area [K] */
32
memory_offset      /* Offset of EL interface into shared memory */
0x00C000
control_register0    /* Value of Control register 0 on initialization */
0xC0
control_register1    /* Value of Control register 1 on initialization */
0x0
control_register2    /* Value of Control register 2 on initialization */
0x0040
clear_shared_mem_at_start  /* "yes" or "no" */
yes
/*****************************************************************************/
```